The Inherent Time Complexity and An Efficient Algorithm for Subsequence Matching Problem

Zemin Chao
Harbin Institute of Technology
Harbin, China
Shenzhen Institute of Advanced Technology Chinese Academy of Sciences
Shenzhen, China
chaozm@hit.edu.cn

Hong Gao
Harbin Institute of Technology
Harbin, China
honggao@hit.edu.cn

Yinan An
Harbin Institute of Technology
Harbin, China
21S003037@stu.hit.edu.cn

Jianzhong Li
Shenzhen Institute of Advanced Technology Chinese Academy of Sciences
Shenzhen, China
Harbin Institute of Technology
Harbin, China
lijzh@hit.edu.cn

ABSTRACT

Subsequence matching is an important and fundamental problem on time series data. This paper studies the inherent time complexity of the subsequence matching problem and designs a more efficient algorithm for solving the problem. Firstly, it is proved that the subsequence matching problem is incomputable in time $O(n^{1-\delta})$ even allowing polynomial time preprocessing if the hypothesis SETH is true, where n is the size of the input time series and $0 \leq \delta < 1$, i.e. , the inherent complexity of the subsequence matching problem is $\omega(n^{1-\delta})$. Secondly, an efficient algorithm for subsequence matching problem is proposed. In order to improve the efficiency of the algorithm, we design a new summarization method as well as a novel index for series data. The proposed algorithm supports both Euclidean Distance and DTW distance with or without z-normalization. Experimental results show that the proposed algorithm is up to about $3 \sim 10$ times faster than the state of art algorithm on the constrained z-normalized Euclidean Distance and DTW distance, and is up to $7 \sim 12$ times faster on Euclidean Distance.

PVLDB Reference Format:
Zemin Chao, Hong Gao, Yinan An, and Jianzhong Li. The Inherent Time Complexity and An Efficient Algorithm for Subsequence Matching Problem. PVLDB, 15(7): 1453 - 1465, 2022.
doi:10.14778/3523210.3523222

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://gitee.com/su_wen_chang/subsequence_matching.

Proceedings of the VLDB Endowment, Vol. 15, No. 7 ISSN 2150-8097.
doi:10.14778/3523210.3523222

1 INTRODUCTION

Time series data are common in industry and daily life, such as wind speed collected by sensors, prices of stocks, and body movements measured by wearable devices. All these information can be abstracted as a series of real numbers representing the changing data with the elapse of time. We assume that the data arrives at equal time intervals in this paper. Therefore, a time series S of length n is formalized as an ordered sequence of n real numbers, i.e. $S = (s_1, s_2, s_3, ...s_n)$. An m-length subsequence of S is a sequence defined as $S_i^m = (s_i, s_{i+1}, s_{i+2}, ...s_{i+m-1})$, where $1 \leq i \leq n - m + 1$ and $m > 0$.

There are two important types of similarity search on series data. One is *whole sequence matching*, which is to find the similar series of the query in a set of series [7, 25, 30].

The other is *subsequence matching*, which is one of the most important but time consuming operation on time series data. This paper focuses on the *subsequence matching problem*.

Subsequence matching problem is frequently involved in many applications such as motif discovery, anomaly detection and financial analysis [2, 12, 23, 31–34]. It takes a data series $S = (s_1, s_2, ..., s_n)$, a query series $Q = (q_1, q_2, ..., q_m)$ and threshold $\epsilon > 0$ as its input, and outputs the result $ANS(S, Q, \epsilon) = \{X | X = S_i^{|Q|} \wedge D(X, Q) \leq \epsilon, 1 \leq i \leq 1, 2, ..., n - m + 1\}$, where $D(X, Q)$ is a distance measurement defined on X and Q.

The *subsequence matching problem* has attracted lots of research interests. The first algorithm for *subsequence matching* on Euclidean Distance is proposed in [8]. Then, other two algorithms, called Dual Match and General Match, are proposed with improved segmentation strategy [17, 18]. Zhu et al. propose envelope technics to deal with DTW distance [33]. Further more, multiple distance measurements are supported in [9]. In addition, there are also hash based approximate solutions to improve the query efficiency [1]. Besides, Edit Distance and its variants are also used in subtrajectory similarity search [5, 6, 15, 20, 24, 26].

The above efforts focused on searching subsequences on raw series. Recently, the importance to support subsequence matching between z-normalized subsequences is recognized [16, 19, 29]. The z-normalized form of series $X = (x_1, x_2, ..., x_m)$, denoted by $\hat{X}$, is

$$\hat{X} = (\frac{x_1 - \mu_X}{\sigma_X}, \frac{x_2 - \mu_X}{\sigma_X}, ..., \frac{x_m - \mu_X}{\sigma_X}),$$

where $\mu_X = \frac{1}{m}\sum_{i=1}^{m} x_i$ and $\sigma_X = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\mu_X - x_i)^2}$.

Z-normalization is necessary for recognizing similar series with tolerance for vertical scaling and horizontal shifting [10]. For example, two stocks with similar trend may have significant difference in their prices. Therefore, directly applying distance functions on the raw subsequences would not be able to reasonably reveal their similarity.

Considering two $|Q|$-length subsequences of S, X and $X^{'}$, which are illustrated in Figure 1. Although the shape of Q is more similar to X rather than $X^{'}$, Euclidean Distance or DTW would report that compared with X, subsequence $X^{'}$ is more similar to Q. Therefore, sometimes it is not only reasonable but also essential to normalize Q, X and $X^{'}$ into $\hat{Q}$, $\hat{X}$ and $\hat{X^{'}}$ first, and then comparing $D(\hat{Q}, \hat{X})$ and $D(\hat{Q}, \hat{X^{'}})$. Z-normalization is useful, but the normalized series are incomputable in preprocessing without the knowledge of $|Q|$, which brings troubles to index-based methods. Therefore, there are only a few algorithms support for Z-normalization.

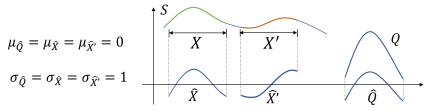


Figure 1: An illustrative example of subsequence matching.

UCR suite [19] provides the first practical solution to the subsequence matching problem on z-normalized subsequences. However, it requires to scan S for each query. To the best of our knowledge, KV-Match [29] and ULISSE [16] are the state-of-art approaches that support z-normalization without the need for scanning the whole series.

However, ULISSE has two drawbacks. The first one is that ULISSE requires $l_{min} \leq |Q| \leq l_{max}$, where l_{min} and l_{max} are predefined constants. Therefore, it cannot handle query series of arbitrary lengths. The second one is that the results of the z-normalization may not satisfy the user's requirements in some applications, such as IoT. As indicated in [29], because z-normalization completely eliminates the differences among the means and standard deviations of series, which may carry important information for identifying the series of interest.

The KV-match algorithm proposes to add a constraint for subsequence matching and overcomes the above drawbacks. Specifically, the constraint is denoted as $R_{cons} = [\beta_1, \beta_2] \times [\alpha_1, \alpha_2]$, $\{\alpha_1^{'}, \alpha_2^{'}, \beta_1^{'}, \beta_2^{'}\} \subset \mathbb{R}$, which represents a legal "rectangle" range of the mean and standard deviation for any $X \in ANS(S, Q, \epsilon)$. It is required that the mean of X, denoted by μ_X, and the standard deviation of X, denoted by σ_X, must satisfy $(\mu_x, \sigma_x) \in R_{cons}$, i.e. ,$\mu_X \in [\beta_1, \beta_2]$ and $\sigma_X \in [\alpha_1, \alpha_2]$.

However, the efficiency of KV-match is sensitive to R_{cons}. As shown by the experiments in [29], the time cost varies more than 20 times with different constraint R_{cons}. Actually, KV-Match is not efficient if the "size" of R_{cons}, which is $(\beta_2 - \beta_1) \times (\alpha_2 - \alpha_1)$, is big.

To overcome the disadvantage of KV-Match, this paper proposes a more efficient algorithm to reduce the effect of relative big R_{cons} by using a new index. This algorithm consists of the following three phases.

Phase one calculates a candidate set based on the proposed summarization method and R_{cons}.

Phase two estimates a rough range of μ_X and σ_X according to the index, which is denoted by $R_X = [\beta_1^{'}, \beta_2^{'}] \times [\alpha_1^{'}, \alpha_2^{'}]$ for each candidate subsequences X. Then, the candidate subsequences are filtered according to the new constraint $R_{cons} \cap R_X$ rather than R_{cons}.

Phase three is to simply verify the candidates by retrieving data from S, and output $ANS(S, Q, \epsilon)$.

Note that R_{cons} is an input of KV-Match and could be big. Unfortunately, the size of R_X is much smaller than the size of R_{cons} for any X since R_X only relies on the index. Thus, the proposed algorithm is much more efficient. In addition, the estimation on the lower bound of $D(X, Q)$ is tighter based on the new summarization method, and further reduces the number of candidates.

Besides, the inherent complexity of the *subsequence matching problem* has been proved. To the best of our knowledge, this is the first work on the inherent complexity of the *subsequence matching problem*.

The major contributions of this paper are as follows.

(1) The inherent time complexity, $\omega(n^{1-\delta})$, of the *subsequence matching problem* is proved for the first time, that is, *subsequence matching problem* is not computable in $O(n^{1-\delta})$ even with any polynomial time preprocessing if SETH [1] is true, where $\delta \in (0, 1)$ and n is the length of S.
(2) A new summarization method, called *Extended Piecewise Aggregate Approximation* (*EPAA*), is proposed. *EPAA* has the following advantages: 1) *EPAA* provides an upper bound for Euclidean Distance as well as *DTW*, which can deal with both nearest and furthest neighbor queries with the same data structure, and 2) *EPAA* supports the estimation of R_X, which improves the efficiency of solving subsequence matching problem.
(3) A new index is proposed based on *EPAA* summarization. In addition, an efficient algorithm for solving the *subsequence matching problem* is proposed, which uses the lower bounding functions and cascading filter strategy proposed in this paper.
(4) Experimental results show that *EPAA* is able to provide high quality distance lower bound of z-normalized Euclidean Distance and z-normalized *DTW* Distance. Besides, the proposed algorithm is up to $3 \sim 10$ times faster than KV-match on the constrained z-normalized *ED* and *DTW* distance, and is up to $7 \sim 12$ times faster than KV-match on Euclidean Distance.

The rest of this paper is organized as follows. Section II introduces related preliminaries, defines the problem, and analyzes the

[1]Let $s_k \in \mathbb{R}$ to be the infimum of δ that *k-SAT* problem can be solved in $O(2^{\delta n})$ time, where n is the number of variables in the given *k-SAT* instance. Strong Exponential Time Hypothesis (*SETH*) is the conjecture that $s_3 > 0$ and $s_\infty = 1$ [11].

inherent complexity of the problem. Section III presents the *EPAA* method and the related lower bounding functions. Section IV discusses a new index based on *EPAA* and the proposed algorithm for subsequence matching problem in details. Section V evaluates *EPAA* summarization mathod and compares the proposed algorithm with other state-of-art algorithms by experiments. Finally, Section VI concludes the paper.

For easy to read, some frequently used symbols in the rest of the paper are listed in Table 1.

Table 1: Frequently Used Notations

Symbol	Description
S	the long series to be searched.
Q	the query series.
S_i^m	the m-length subsequence of S that starts from the ith position.
μ_X, σ_X	the mean and standard deviation of series X.
U, L	envelopes defined by LB_{keogh} [14].
$ANS(S, Q, \epsilon)$	the result of subsequence matching problem under input S, Q and ϵ.

2 PROBLEM DEFINITION AND COMPLEXITY

2.1 Preliminaries and Problem Definition

The focus of this paper is to solve the *subsequence matching problem* for the most common distance measurements in the industry and finance applications. Let m be the length of query series Q and X be an m-length subsequence of S. This paper supports for the following four distance measurements defined on X and Q. The first is the Euclidean Distance, which is defined as

$$ED(X, Q) = \sqrt{\sum_{i=1}^{m}(x_i - q_i)^2}.$$

The second is the Dynamic Time Warping (DTW). DTW allows two series to align with each other before measuring distance point by point, which is recursively defined as

$$DTW(X, Q) = \sqrt{(x_1 - q_1)^2 + MIN},$$

where $MIN = min\{DTW(X_2^{m-1}, Q_2^{m-1}), DTW(X_2^{m-1}, Q), DTW(X, Q_2^{m-1})\}$, and the distance between arbitrary sequence and an empty series is defined as ∞.

In practice, this alignment is restricted by bands to prevent the pathological warping path. In the rest of the paper, we assume Sakoe-Chiba band [22] is used. The choice of band has no effect on the proposed method.

Given $R_{cons} = [\beta_1, \beta_2] \times [\alpha_1, \alpha_2]$, X and Q, let $\hat{X}$ and $\hat{Q}$ be the z-normalized series of X and Q. The R_{cons}-constrained z-normalized Euclidean Distance($CNED$) between X and Q is defined as

$$CNED(X, Q) = \begin{cases} ED(\hat{X}, \hat{Q}), & if \quad (\mu_X, \sigma_X) \in R_{cons} \\ \infty, & otherwise. \end{cases}$$

Given $R_{cons} = [\beta_1, \beta_2] \times [\alpha_1, \alpha_2]$, X and Q, the R_{cons}-constrained and z-normalized DTW distance between X and Q is defined as

$$CNDTW(X, Q) = \begin{cases} DTW(\hat{X}, \hat{Q}), & if \quad (\mu_X, \sigma_X) \in R_{cons} \\ \infty, & otherwise. \end{cases}$$

Without loss of generality, we can reasonably assume that $\hat{Q}$=Q in CNED or CNDTW queries, otherwise we can simply replace Q with $\hat{Q}$. The *subsequence matching problem* is defined by the following Definition 1.

Definition 1. *[Subsequence Matching Problem] Given a series S, a query series Q, a distance measurement D, and $\epsilon > 0$, the subsequence matching problem is to find the set $ANS(S, Q, \epsilon) = \{X | \exists i, X = S_i^{|Q|}, D(X, Q) \leq \epsilon\}$.*

2.2 Inherent Complexity of Subsequence Matching Problem

Although *subsequence matching problem* has been studied for years, there has been no analysis on its inherent complexity till now. Our result is formalized as Theorem 1 and the detailed proof is in the Appendix of the paper.

Theorem 1. *Let $n = |S|$. If Strong Exponential Time Hypothesis is true, then the inherent time complexity of subsequences matching problem is $\omega(n^{1-\delta})$ under any L_p metric for $p > 0$ even if S has been preprocessed in time $O(n^c)$ for any constant $c > 1$ and $0 < \delta < 1$.*

Corollary 1. *The inherent time complexity of subsequences matching problem under DTW with Sakoe-Chiba band, is also $\omega(n^{1-\delta})$.*

Proof. L_2 norm is a special case of DTW with $r = 0$ (Sakoe-Chiba band). Therefore, *subsequences matching problem* under DTW is not computable in $O(n^{1-\delta})$ time due to Theorem 1. □

Theorem 1 indicates that even if we leverage index to accelerate the query processing, any practicable algorithm for *subsequence matching problem* under any L_p measurement, still requires $\omega(|S|^{1-\delta})$ time theoretically. Considering the brute-force algorithm costs $O(|Q||S|)$ time and $|Q| << |S|$, the gap between brute-force algorithm and the theoretical lower bound of the complexity is rather small. This explains why all the existing researches on this issue as well as this paper cannot improve the complexity of algorithms.

Our result means that it is unlikely to find any practicable algorithm with time complexity significantly less than $\Theta(n)$. Therefore, the filter-verify framework is perhaps the best solution we can expect for *subsequence matching problem.* Although we prove that indexes cannot be helpful to improve the time complexity of the algorithm, carefully arranged filtering strategy and index, which are the focus of the following sections, are still able to significantly accelerate the query in practice.

3 MATHEMATICAL FOUNDATION

Summarization methods summarize a series X as $\widetilde{X}$, such that the lower bound of $D(X, Q)$ can be computed by some function $D_{lb}(\widetilde{X}, Q)$ for any Q such that if $D_{lb}(\widetilde{X}, Q) > \epsilon$ then $D(X, Q) > \epsilon$ and $X \notin ANS(S, Q, \epsilon)$. Therefore, X can be pruned without accessing the series X itself.

This section presents the mathematical foundation of the proposed *EPAA* method. Firstly, subsection 3.1 presents how to compute $\widetilde{X}$ in *EPAA* summarization method. Then, subsections 3.2-3.3 present the lower bound functions of *EPAA* method, $D_{lb}(\widetilde{X}, Q)$, for *ED*, *DTW*, *CNED* and *CNDTW* respectively. Finally, subsection 3.4 discusses how to leverage *EPAA* to quickly filter out non-candidate subsequences without directly calculating $D_{lb}(\widetilde{X}, Q)$.

3.1 Summarizing Series with EPAA

The existing summarization methods use a linear combination of elements in X to represent X for supporting queries on both *ED* and *DTW*. Unfortunately, they cannot be used to estimate R_X in *subsequence matching problem* such that the normalized queries cannot be efficiently processed.

To solve this problem, we propose a new summarization method, called Extended Piecewise Aggregate Approximation (*EPAA*), which extends Piecewise Aggregate Approximation (*PAA*) [13] by adding the standard deviation of each segments.

For simplicity, in the rest of the paper, we say a series X is *completely decomposed into t continuous disjoint segments* if X is divided to segments $\{X_{p_i}^{w_i}\}$ for $i \in \{1, 2, ..., t\}$, where $p_1 = 1$, $w_i, t \in \mathbb{N}^+$, $p_i + w_i = p_{i+1}$ and $\sum_{i=1}^{t} w_i = |X|$. Suppose a series X is *completely decomposed into t continuous disjoint segments* $\{X_{p_i}^{w_i}\}$, *EPAA* method calculates $\widetilde{X}$ as

$$\widetilde{X} = (\mu_{X_{p_1}^{w_1}}, \sigma_{X_{p_1}^{w_1}}, \mu_{X_{p_2}^{w_2}}, \sigma_{X_{p_2}^{w_2}}, ..., \mu_{X_{p_t}^{w_t}}, \sigma_{X_{p_t}^{w_t}}),$$

where $\mu_{X_{p_i}^{w_i}}$ and $\sigma_{X_{p_i}^{w_i}}$ are the mean and standard deviation of the ith segment for $i \in \{1, ..., t\}$. $\widetilde{X}$ will be used to calculate lower bounds for *ED*, *DTW*, *CNED* and *CNDTW*.

In fact, *EPAA* provides an alternative summarization method for any problem involving series data summarization, and is not restricted to *subsequence matching problem.*

3.2 The Lower Bounds of DTW and CNDTW

This subsection gives the details of $D_{lb}(\widetilde{X}, Q)$ for DTW and CNDTW, which are denoted by $DTW_{lb}(\widetilde{X}, Q)$ and $CNDTW_{lb}(\widetilde{X}, Q, R_{cons})$ respectively.

3.2.1 Lower Bound of DTW. First, when the number of segments in X being one, i.e. $\widetilde{X} = (\mu_X, \sigma_X)$, the lower bound of $DTW(X, Q)$ is determined by the following Lemma 1.

Lemma 1. *Suppose r is the parameter of Sakoe-Chiba band [22]. Let* $L = (l_1, l_2, ..., l_m)$ *and* $U = (u_1, u_2, ..., u_m)$, *where* $u_i = max\{q_{i-r}, q_{i-r+1}, ..., q_{i+r}\}$ *and* $l_i = min\{q_{i-r}, q_{i-r+1}, ..., q_{i+r}\}$. *Given two m-length series X and Q, if* $ED_{lb}(\widetilde{X}, L)^2 + ED_{lb}(\widetilde{X}, U)^2 \geq ED(L, U)^2$, *then there is a lower bound of* $DTW(X, Q)$ *that is*

$$DTW_{lb}(\widetilde{X}, U, L) = \frac{1}{2}[-ED(L, U) + \sqrt{2ED_{lb}(\widetilde{X}, L)^2 + 2ED_{lb}(\widetilde{X}, U)^2 - ED(L, U)^2}]. \quad (1)$$

Proof. A well-known lower bound of $DTW(X, Q)$ is LB_{Keogh}[14], which is,

$$LB_{Keogh}(X, Q) = \sqrt{\sum_{i=1}^{m} \begin{cases} (u_i - x_i)^2, & x_i \geq u_i \\ (l_i - x_i)^2, & x_i \leq l_i \\ 0, & l_i \leq x_i \leq u_i. \end{cases}}$$

Let

$$a_i = \begin{cases} |x_i - l_i|, & x_i \leq l_i \\ |u_i - x_i|, & others, \end{cases}$$

and $b_i = |u_i - l_i|, i \in \{1, 2....m\}$, then

$$\begin{aligned} ED(X, L)^2 + ED(X, U)^2 &= \sum_{i=1}^{m} (l_i - x_i)^2 + (u_i - x_i)^2 \\ &= \sum_{x_i \leq l_i \ or \ u_i \leq x_i} (2a_i^2 + 2a_i b_i + b_i^2) + \sum_{l_i \leq x_i \leq u_i} (2a_i^2 - 2a_i b_i + b_i^2) \\ &= \sum_{x_i \leq l_i \ or \ u_i \leq x_i} (2a_i^2 + 2a_i b_i) + \sum_{l_i \leq x_i \leq u_i} 2a_i(a_i - b_i) + \sum_{i=1}^{m} (b_i^2). \end{aligned}$$

Since $a_i \leq b_i$ if $l_i \leq x_i \leq u_i$,

$$ED(X, L)^2 + ED(X, U)^2 \leq \sum_{x_i \leq l_i \ or \ u_i \leq x_i} (2a_i^2 + 2a_i b_i) + \sum_{i=1}^{m} (b_i^2). \quad (2)$$

By Cauchy- Schwarz inequality, we have

$$\begin{aligned} \sum_{x_i \leq l_i \ or \ u_i \leq x_i} a_i b_i &\leq \sqrt{\sum_{x_i \leq l_i \ or \ u_i \leq x_i} (a_i^2) * \sum_{x_i \leq l_i \ or \ u_i \leq x_i} (b_i^2)} \\ &\leq LB_{Keogh}(X, Q) \sqrt{\sum_{i=1}^{m} (b_i^2)}. \end{aligned} \quad (3)$$

Combining formula (2) and formula (3), we have

$$\begin{aligned} &2LB_{Keogh}(X, Q)^2 + 2LB_{Keogh}(X, Q) \sqrt{\sum_{i=1}^{m} (b_i^2)} \\ &+ \sum_{i=1}^{m} (b_i^2) - ED(X, L)^2 - ED(X, U)^2 \geq 0. \end{aligned} \quad (4)$$

Eq (4) is a *quadratic inequality* about $LB_{Keogh}(X, Q)$. Its discriminant $\Delta = 8ED(X, L)^2 + 8ED(X, U)^2 - 4\sum_{i=1}^{m} (b_i^2)$. Since $\sum_{i=1}^{m} (b_i^2) = ED(L, U)^2$, $\Delta \geq 0$ is inducted by applying *Cosine Theorem* on $ED(X, L), ED(X, U)$ and $ED(L, U)$. Considering the fact $LB_{Keogh}(X, Q) \geq 0$, we have

$$LB_{Keogh}(X, Q) \geq \frac{1}{2}[-\sqrt{\sum_{i=1}^{m} (b_i^2)} + \sqrt{2ED(X, L)^2 + 2ED(X, U)^2 - \sum_{i=1}^{m} (b_i^2)}]. \quad (5)$$

Since $LB_{Keogh}(X, Q) \leq DTW(X, Q)$, the right side of Eq.5 is no greater than $DTW(X, Q)$. Therefore, the lemma is true by substituting $ED(X, L)$ and $ED(X, U)$ with their lower bound $ED_{lb}(\widetilde{X}, L)$ and $ED_{lb}(\widetilde{X}, U)$ respectively. □

Now, we extend the conclusion of Lemma 1, to the general case that the number of segments in X is t ($t \geq 1$), where $\widetilde{X} = (\mu_{X_{p_1}^{w_1}}, \sigma_{X_{p_1}^{w_1}}, \mu_{X_{p_2}^{w_2}}, \sigma_{X_{p_2}^{w_2}}, ..., \mu_{X_{p_t}^{w_t}}, \sigma_{X_{p_t}^{w_t}})$.

Theorem 2 (distance lower bound of dtw). *Supposing two m-length series X and Q are divided into t continuous disjoint segments $\{X_{p_i}^{w_i}\}$ and $\{Q_{p_i}^{w_i}\}$, a distance lower bound of $DTW(X,Q)$ is*

$$DTW_{lb}(\widetilde{X}, Q) = \sqrt{\sum_{i=1}^{t} DTW_{lb}(\widetilde{X}_{p_i}^{w_i}, U_{p_i}^{w_i}, L_{p_i}^{w_i})^2}. \tag{6}$$

where $\widetilde{X}_{p_i}^{w_i} = (\mu_{X_{p_i}^{w_i}}, \sigma_{X_{p_i}^{w_i}})$ is the mean and standard deviation of the ith segment in X.

Proof. From the definition of LB_{Keogh}, we have

$$\begin{aligned} DTW(X,Q)^2 &\geq \sum_{i=1}^{m} \begin{cases} (u_i - x_i)^2, & x_i \geq u_i \\ (l_i - x_i)^2, & x_i \leq l_i \\ 0, & l_i \leq x_i \leq u_i \end{cases} \\ &= \sum_{j=1}^{t} \sum_{k=p_j}^{p_{j+1}-1} \begin{cases} (u_k - x_k)^2, & x_k \geq u_k \\ (l_k - x_k)^2, & x_k \leq l_k \\ 0, & l_k \leq x_k \leq u_k \end{cases} \\ &\geq \sum_{i=1}^{t} DTW_{lb}(\widetilde{X}_{p_i}^{w_i}, U_{p_i}^{w_i}, L_{p_i}^{w_i})^2 \end{aligned}$$

□

3.2.2 Lower Bound of CNDTW. If the number of segments in X is one, we have Lemma 2.

Lemma 2. *Given $R_{cons} = [\beta_1, \beta_2] \times [\alpha_1, \alpha_2]$, two m-length series X and Q, if $a + b - ED(L,U)^2 \geq 0$, there is a lower bound of $CNDTW(X,Q)$ which is*

$$CNDTW_{lb}(\widetilde{X}, U, L, R_{cons}) = \frac{1}{2}[-ED(L,U) + \sqrt{2a + 2b - ED(L,U)^2}], \tag{7}$$

where

$$a = m \times \min_{\mu_{\hat{X}} \in [\hat{\mu}_{min}, \hat{\mu}_{max}]} \{2\mu_{\hat{X}}^2 - 2(\mu_L + \mu_U)\mu_{\hat{X}} + \mu_L^2 + \mu_U^2\}$$

$$b = m \times \min_{\sigma_{\hat{X}} \in [\hat{\sigma}_{min}, \hat{\sigma}_{max}]} \{2\sigma_{\hat{X}}^2 - 2(\sigma_L + \sigma_U)\sigma_{\hat{X}} + \sigma_L^2 + \sigma_U^2\}$$

$$\hat{\mu}_{min} = min\{\frac{\mu_{min} - \beta_2}{\alpha_1}, \frac{\mu_{min} - \beta_2}{\alpha_2}\}, \hat{\sigma}_{min} = \frac{\sigma_{min}}{\alpha_2}$$

$$\hat{\mu}_{max} = max\{\frac{\mu_{max} - \beta_1}{\alpha_1}, \frac{\mu_{max} - \beta_1}{\alpha_2}\}, \hat{\sigma}_{max} = \frac{\sigma_{max}}{\alpha_1}.$$

Proof. (sketch) Denoting the summarization of constrained z-normalized series $\hat{X}$ by $\widetilde{\hat{X}}$, CNDTW means applying DTW on z-normalized series $\hat{X}$ instead of raw series X. Therefore, $DTW_{lb}(\widetilde{\hat{X}}, Q) \leq CNDTW(X,Q)$. Given $\widetilde{X} \in [\mu_{min}, \mu_{max}] \times [\sigma_{min}, \sigma_{max}]$ and $R_{cons} = [\beta_1, \beta_2] \times [\alpha_1, \alpha_2]$, it can be proved that $\widetilde{\hat{X}} \in [\hat{\mu}_{min}, \hat{\mu}_{max}] \times [\hat{\sigma}_{min}, \hat{\sigma}_{max}]$. Since $DTW_{lb}(\widetilde{\hat{X}}, Q)$ is a continuous function of $\hat{X}$ if $2ED(\widetilde{\hat{X}}, L)^2 + 2ED(\widetilde{\hat{X}}, U)^2 - ED(L,U)^2 \geq 0$, its minimum value can be obtained as Eq.7. □

Extending the conclusion of Lemma 2, we have the lower bound of $CNDTW$ in general case of there bing t segments in X.

Theorem 3 (distance lower bound of cndtw). *Supposing two m-length series X and Q are divided into t continuous disjoint segments $\{X_{p_i}^{w_i}\}$ and $\{Q_{p_i}^{w_i}\}$, there is a lower bound of $CNDTW(X,Q)$ which is*

$$CNDTW_{lb}(\widetilde{X}, Q, R_{cons}) = \sqrt{\sum_{i=1}^{t} CNDTW_{lb}(\widetilde{X}_{p_i}^{w_i}, U_{p_i}^{w_i}, L_{p_i}^{w_i}, R_{cons})^2}. \tag{8}$$

Proof. (sketch) Because $CNDTW$ can be seen as applying DTW to the z-normalized subsequences, the proof can be done by applying the similar approach in the proof of Theorem 2 on the z-normalized subsequences. □

3.3 The Lower Bounds of ED and CNED

This subsection presents $D_{lb}(\widetilde{X}, Q)$ for ED and CNED, which are denoted as $ED_{lb}(\widetilde{X}, Q)$ and $CNED_{lb}(\widetilde{X}, Q, R_{cons})$.

3.3.1 Lower Bound of ED. Let X and Q be two m-length series. When the number of segment is one, a lower bound of $ED(X,Q)$ is as

$$ED_{lb}(\widetilde{X}, Q) = \sqrt{m \times [(\mu_X - \mu_Q)^2 + (\sigma_X - \sigma_Q)^2]}, \tag{9}$$

which has been proven in [10] and [28].

Theorem 4 (distance lower bound of ED). *Suppose two m-length series X and Q are divided into t continuous disjoint segments $\{X_{p_i}^{w_i}\}$ and $\{Q_{p_i}^{w_i}\}$, there is a lower bound of $ED(X,Q)$, which is*

$$ED_{lb}(\widetilde{X}, Q) = \sqrt{\sum_{i=1}^{t} ED_{lb}(\widetilde{X}_{p_i}^{w_i}, Q_{p_i}^{w_i})^2}. \tag{10}$$

3.3.2 Lower Bound of CNED. In case that there is only one segment in the series, we have Lemma 3.

Lemma 3. *Given $R_{cons} = [\beta_1, \beta_2] \times [\alpha_1, \alpha_2]$, two m-length series X and Q, there is a lower bound of $CNED(X,Q)$, which is*

$$CNED_{lb}(\widetilde{X}, Q, R_{cons}) = \sqrt{c^2 + d^2}. \tag{11}$$

where $\hat{\mu}_{min}, \hat{\sigma}_{min}, \hat{\mu}_{max}, \hat{\sigma}_{max}$ are the same as Lemma 2, and

$$c = \begin{cases} 0, & \mu_Q \in [\hat{\mu}_{min}, \hat{\mu}_{max}] \\ m \times min\{(\hat{\mu}_{min} - \mu_Q)^2, (\hat{\mu}_{max} - \mu_Q)^2\}, others \end{cases}$$

$$d = \begin{cases} 0, & \sigma_Q \in [\hat{\sigma}_{min}, \hat{\sigma}_{max}] \\ m \times min\{(\hat{\sigma}_{min} - \sigma_Q)^2, (\hat{\sigma}_{max} - \sigma_Q)^2\}, others \end{cases}$$

Proof. (sketch) The proof is similar to that of Theorem 3, and can be completed by estimating the range of $\widetilde{\hat{X}}$ and finding the minimum value of $ED_{lb}(\widetilde{\hat{X}}, Q)$. □

Extending Lemma 3, we can derive the lower bound of $CNED$ when the number of segments is t.

Theorem 5 (distance lower bound of CNED). *Supposing that two m-length series X and Q are divided into t continuous disjoint segments $\{X_{p_i}^{w_i}\}$ and $\{Q_{p_i}^{w_i}\}$, we have,*

$$CNED_{lb}(\widetilde{X}, Q, R_{cons}) = \sqrt{\sum_{i=1}^{t} CNED_{lb}(\widetilde{X}_{p_i}^{w_i}, Q_{p_i}^{w_i}, R_{cons})^2}. \quad (12)$$

Proof. Because $CNED$ can be seen as a special case of $CNDTW$, i.e. $r = 0$, the theorem is true according to Theorem 3. □

3.4 Fast Filter Based on *EPAA* method

Since calculating $D_{lb}(\widetilde{X}, Q)$ for every subsequence in S could be expensive, this subsection discusses how to quickly select the candidates with *EPAA* method. Our strategy is to find the necessary conditions of $D_{lb}(\widetilde{X}, Q) \le \epsilon$, which are referred as *filter conditions* in the rest of paper. Specifically, given Q and ϵ, we want to find $\{F_1, F_2, ..., F_t\}$ such that $\widetilde{X}_{p_i}^{w_i} = (\mu_{X_{p_i}^{w_i}}, \sigma_{X_{p_i}^{w_i}}) \in F_i$, for any $1 \le i \le t$ and $X \in ANS(S, Q, \epsilon)$. After that, we are able to filter out the non-candidates swiftly by checking any segment of it. If there exists $1 \le i \le t$ that $\widetilde{X}_{p_i}^{w_i} \notin F_i$, X can be filtered out safely without directly calculating $D_{lb}(\widetilde{X}, Q)$.

The filter conditions of ED, DTW, $CNED$ and $CNDTW$ are presented in Corollary 2, 3, 4 and 5 respectively. They can be derived by manipulating formula (4), (7), (9) and (11). The details of the proofs are omitted for simplicity.

Corollary 2 (filter condition for ED). *For any series X and Q that satisfy $ED(X, Q) < \epsilon$, the following inequality must be true for $1 \le i \le t$.*

$$\frac{\epsilon^2}{w_i} \ge (\mu_{X_{p_i}^{w_i}} - \mu_{Q_{p_i}^{w_i}})^2 + (\sigma_{X_{p_i}^{w_i}} - \sigma_{Q_{p_i}^{w_i}})^2 \quad (13)$$

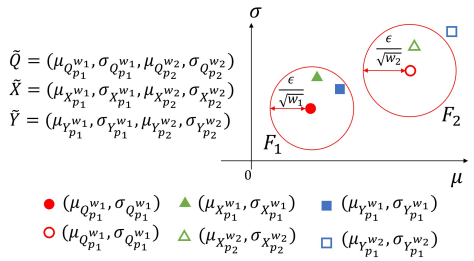


Figure 2: Illustration of F_i for Euclidean Distance.

Regarding $(\mu_{X_{p_i}^{w_i}}, \sigma_{X_{p_i}^{w_i}})$ as the coordinate of a two dimensional point as illustrated in Figure 2, the filter condition F_i corresponding to Euclidean Distance is a disk area centered at $(\mu_{Q_{p_i}^{w_i}}, \sigma_{Q_{p_i}^{w_i}})$ with radius of $\frac{\epsilon}{\sqrt{w_i}}$. Series X satisfies the filter condition since $(\mu_{X_{p_1}^{w_1}}, \sigma_{X_{p_1}^{w_1}}) \in F_1$ and $(\mu_{X_{p_2}^{w_2}}, \sigma_{X_{p_2}^{w_2}}) \in F_2$. Series Y is discarded by F_2 because $(\mu_{Y_{p_2}^{w_2}}, \sigma_{Y_{p_2}^{w_2}}) \notin F_2$.

Corollary 3 (filter condition for DTW). *For any series X and Q that have the same length and $DTW(X, Q) \le \epsilon$, the following inequality must be satisfied for $1 \le i \le t$.*

$$[2\epsilon + ED(L_{p_i}^{w_i}, U_{p_i}^{w_i})]^2 + ED(L_{p_i}^{w_i}, U_{p_i}^{w_i})^2$$
$$\ge 2[ED_{lb}(\widetilde{X}_{p_i}^{w_i}, L_{p_i}^{w_i})^2 + ED_{lb}(\widetilde{X}_{p_i}^{w_i}, U_{p_i}^{w_i})^2].$$

Corollary 4 (filter condition for cned). *If two series X and Q satisfy $CNED(X, Q) \le \epsilon$ and constraint $R_{cons} = [\beta_1, \beta_2] \times [\alpha_1, \alpha_2]$, the following inequalities must be satisfied for $1 \le i \le t$.*

$$\mu_{X_{p_i}^{w_i}} \in [\beta_1 + min\{\alpha_1(\mu_{Q_{p_i}^{w_i}} - \frac{\epsilon}{\sqrt{w_i}}), \alpha_2(\mu_{Q_{p_i}^{w_i}} - \frac{\epsilon}{\sqrt{w_i}})\}$$
$$, \beta_2 + max\{\alpha_1(\mu_{Q_{p_i}^{w_i}} + \frac{\epsilon}{\sqrt{w_i}}), \alpha_2(\mu_{Q_{p_i}^{w_i}} + \frac{\epsilon}{\sqrt{w_i}})\}],$$

where $\mu_{L_{p_i}^{w_i}}, \sigma_{L_{p_i}^{w_i}}, \mu_{U_{p_i}^{w_i}}$ and $\sigma_{U_{p_i}^{w_i}}$ are mean and average of series $L_{p_i}^{w_i}$ and $U_{p_i}^{w_i}$ respectively.

$$\sigma_{X_{p_i}^{w_i}} \in [\alpha_1(\sigma_{Q_{p_i}^{w_i}} - \frac{\epsilon}{\sqrt{w_i}}), max(\alpha_2(\sigma_{Q_{p_i}^{w_i}} + \frac{\epsilon}{\sqrt{w_i}}), \alpha_2\sqrt{\frac{|Q|}{w_i}})]$$

Corollary 5 (filter condition for cndtw). *If two series X and Q satisfy $CNDTW(X, Q) \le \epsilon$ and constraint $R_{cons} = [\beta_1, \beta_2] \times [\alpha_1, \alpha_2]$, the following inequalities must be satisfied for $1 \le i \le t$.*

$$\mu_{\hat{X}_{p_i}^{w_i}} \in [\frac{\mu_{L_{p_i}^{w_i}} + \mu_{U_{p_i}^{w_i}}}{2} - \Delta_1, \frac{\mu_{L_{p_i}^{w_i}} + \mu_{U_{p_i}^{w_i}}}{2} + \Delta_1]$$

$$\sigma_{\hat{X}_{p_i}^{w_i}} \in [\frac{\sigma_{L_{p_i}^{w_i}} + \sigma_{U_{p_i}^{w_i}}}{2} - \Delta_2, min\{\frac{\sigma_{L_{p_i}^{w_i}} + \sigma_{U_{p_i}^{w_i}}}{2} + \Delta_2, \alpha_2\sqrt{\frac{|Q|}{w_i}}\}]$$

where $\Delta_1 = \frac{1}{2}\sqrt{\frac{[2\epsilon + ED(L,U)]^2 + ED(L,U)^2}{w_i} - (\mu_U - \mu_L)^2}$,
$\Delta_2 = \frac{1}{2}\sqrt{\frac{[2\epsilon + ED(L,U)]^2 + ED(L,U)^2}{w_i} - (\sigma_U - \sigma_L)^2}$. $\mu_{\hat{X}_{p_i}^{w_i}}$ *and* $\sigma_{\hat{X}_{p_i}^{w_i}}$ *are the mean and the standard deviation of the ith segments of the normalized series.*

4 SUBSEQUENCE MATCHING ALGORITHM

This section proposes an index according to *EPAA* summarization method. Then, based on the index, a three-phase algorithm to *subsequence matching problem* for ED, DTW, $CNED$ and $CNDTW$ is designed.

4.1 Indexing Series Data

As illustrated in Figure 4(c), the proposed index is a grid structure composed of cells on a two-dimensional plane, where the coordinates are mean and standard deviation of segments in series S respectively. Therefore, each cell in the grid corresponds to a unique combination of an interval of mean and an interval of standard deviation. We use $R(c)$ to denote the range of mean and standard deviation associated with cell c, where $R(c) = [\beta_1', \beta_2'] \times [\alpha_1', \alpha_2']$ and $\{\alpha_1', \alpha_2', \beta_1', \beta_2'\} \subset \mathbb{R}$.

Generally speaking, the index of series S is obtained by mapping w-length segments in S into cells as illustrated in Figure 4(a). Given a predefined segment length w, The mean value and standard deviation of each w-length subsequence in S, which are $\{\mu_{S_i^w}\}$ and

$\{\sigma_{S_i^w}\}$ for $1 \le i \le |S| - w + 1$, are calculated. Then, each segment S_i^w is assigned to the cell c, such that $(\mu_{S_i^w}, \sigma_{S_i^w}) \in R(c)$.

For each cell c, the index records the following information: 1) A set $SEG_c = \{i | (\mu_{S_i^w}, \sigma_{S_i^w}) \in R(c), 1 \le i \le |S|-w+1\}$, which denotes the segments mapped to c by their start positions, and 2) A range $R^{sur}(c) \in [\beta_1', \beta_2'] \times [\alpha_1', \alpha_2']$ such that $\forall i \in SEG_c, (\mu_{S_{i+w}^j}, \sigma_{S_{i+w}^j}) \in R^{sur}(c)$ for any $j \in \{1, 2, ...w-1\}$.

The grid in figure 4(c) is obtained by dividing the rows and columns of the two dimensional plane according to quantiles of $\{\mu_{S_i^w}\}$ and $\{\sigma_{S_i^w}\}$ respectively. To locate the cells on disk, our index includes metadata that contains auxiliary information of cells, such as the range of signatures, its offset on disk, the number of segments contained, etc. The information of the cells are stored on disk in the order of Hilbert curve.

Besides, SEG_c is compressed to save I/O overhead. Consider two w-length segments $S_i^w = (s_i, s_{i+1}, ..., s_{i+w-1})$, and $S_{i+1}^w = (s_{i+1}, s_{i+2}, ..., s_{i+w})$. Their difference is at most one element. Thus S_i^w and S_{i+1}^w are likely to have similar mean and standard deviation and to be mapped into the same cell c consequently. Therefore, we are able to find i and $i+1$ in SEG_c. Further more, many adjacent elements such as $i, i+1, i+2, ..., i+k$ are likely to be found for the same reason. Without loss of generality, we assume that the elements of SEG_c are sorted by ascending order. If $k \ge 2$, $i-1 \notin SEG_c$ and $i+k+1 \notin SEG_c$, we compress $i, i+1, i+2, ..., i+k$ as three sequential elements $\#, i, i+k$, where $\#$ is a special identifier indicating that the following two elements represent a compressed sequence of adjacent elements.

4.2 The Proposed Algorithm

Now we present the algorithm for solving *subsequence matching problem* based on cascading filter strategy, which consists of three phases. The details of the proposed algorithm for *subsequence matching problem* is shown in Algorithm 1.

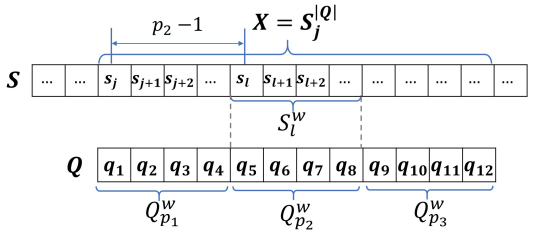


Figure 3: The candidate subsequence, $S_j^{|Q|}$, is $p_i - 1$ ahead of the corresponding candidate segment S_l^w, if $l \in c$ and $c \in C_i$.

Phase I. Index Probing (Lines 1-8). The query series Q is decomposed into t disjoint w-length segments $Q_{p_1}^w, Q_{p_2}^w, ..., Q_{p_t}^w$ in Lines 1-2. Let F_i be the region that corresponds to filter condition derived from $Q_{p_i}^w$ and ϵ. C_i is the set of the cells that satisfy the filter condition F_i. The candidate segments are fetched by loading C_i into memory for every $i \in \{1, 2, ...t\}$ in Lines 3-5.

If $l \in c$ and $c \in C_i$, then segment S_l^w satisfies the filter condition derived from $Q_{p_i}^w$ and ϵ. As illustrated in Figure 3, the candidate subsequence corresponding to element l is $S_{l-p_i+1}^{|Q|}$. Finally, the start

Algorithm 1: SubsequenceMatching(S, Q, ϵ)

Input: Data series $S \in \mathbb{R}^n$, query series $Q \in \mathbb{R}^m$, distance threshold $\epsilon > 0$
Output: $ANS(S, Q, \epsilon)$

1 $CS \leftarrow \emptyset, ANS \leftarrow \emptyset, t \leftarrow \lfloor |Q|/w \rfloor, CS_i \leftarrow \emptyset$ for $1 \le i \le t$;
2 $\{Q_{p_1}^w, Q_{p_2}^w, ..., Q_{p_t}^w\} \leftarrow$ Select t disjoint segments from Q ;
3 **for** $1 \le i \le t$ **do**
4 $\quad F_i \leftarrow$ calculates filter condition by $Q_{p_i}^w$ and ϵ ;
5 $\quad C_i \leftarrow \{c | R(c) \cap F_i \ne \emptyset\}$; // the cells to fetch
6 $\quad$ **for** $c \in C_i$ **do**
7 $\quad\quad$ **for** $l \in SEG_c$ **do**
8 $\quad\quad\quad CS_i \leftarrow CS_i \cup \{< l - p_i + 1, c.ID >\}$;
9 **for** $l \in \{l | \forall i \in \{1, 2..., t\}, \exists c_i \; that < l, c_i > \in CS_i\}$ **do**
10 $\quad$ **for** $1 \le i \le t$ **do**
11 $\quad\quad lb_i \leftarrow \min_{\forall \widetilde{Y} \in R(c_i)} D_{lb}(\widetilde{Y}, Q_{p_i}^w)$; // shared by segments in c_i
12 $\quad$ **if** $\sqrt{\sum_{i=1}^t lb_i^2} \le \epsilon$ **then** // checking $D_{lb}(\widetilde{X}, Q)$
13 $\quad\quad$ **if** *Distance measurement is ED or DTW* **then**
14 $\quad\quad\quad CS \leftarrow CS \cup \{l\}$;
15 $\quad\quad$ **if** *Distance measurement is CNED or CNDTW* **then**
16 $\quad\quad\quad R_X \leftarrow ESTIMATE\_R_X(c_1, c_2, ..., c_t)$ // Algorithm 2 ;
17 $\quad\quad\quad$ **if** $R_X \cap R_{cons} \ne \emptyset$ **then**
18 $\quad\quad\quad\quad$ **for** $1 \le i \le t$ **do**
19 $\quad\quad\quad\quad\quad lb_i \leftarrow \min_{\forall \widetilde{Y} \in R(c_i)} D_{lb}(\widetilde{Y}, Q_{p_i}^w, R_X \cap R_{cons})$;
20 $\quad\quad\quad\quad$ **if** $\sqrt{\sum_{i=1}^t lb_i^2} \le \epsilon$ **then** // $D_{lb}(\widetilde{X}, Q, R_X \cap R_{cons})$
21 $\quad\quad\quad\quad\quad CS \leftarrow CS \cup \{l\}$;
22 **for** $l \in CS$ **do**
23 $\quad$ **if** $D(S_l^{|Q|}, Q) \le \epsilon$ **then** // verify candidates
24 $\quad\quad ANS \leftarrow ANS \cup \{S_l^{|Q|}\}$
25 **return** ANS

positions of the candidate subsequences $S_{l-p_i+1}^{|Q|}$ and the ID of the cell c that contains segment S_l^w is added to CS_i in Lines 6-8.

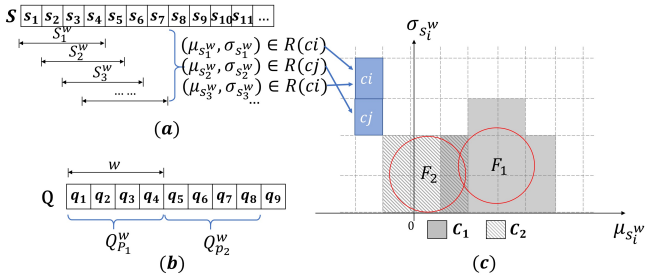


Figure 4: An instance of the proposed index and querying processing. (a) Mapping the segments in S_i^w to cells. (b) Dividing Q into segments. (c) Fetching C_i from the index under Euclidean Distance.

Figure 4 illustrates an example of *phase I*. Assuming that there exist two segments of Q illustrated as Figure 4(b) and the filler conditions are F_1 and F_2 respectively. According to Corollary 2, the cells that satisfy the filter condition are marked as C_1 and C_2 on the grid structure of index as shown in Figure 4(c). Then CS_1 and CS_2 are generated at the end of *Phase I*.

Phase II. Candidate Refining (Lines 9-21). In this phase, we prune the candidate subsequences $\{l\}$ by merging CS_i and checking whether l appears in every CS_i for $1 \le i \le t$. Then the candidate subsequences are further pruned by comparing $D_{lb}(\widetilde{X}, Q)$ with ϵ. In Lines 10-12, $D_{lb}(\widetilde{X}, Q)$ is computed according to $c_1, c_2, ..., c_t$, which are the cells containing $X^w_{p_1}, X^w_{p_2}, ..., X^w_{p_t}$ respectively. The distance contributed by the ith segment of candidate subsequence X is computed as $lb_i = \min_{\forall \widetilde{Y} \in R(c_i)} D_{lb}(\widetilde{Y}, Q^w_{p_i})$. Therefore, lb_i can be shared by all segments in cell c_i. Then, $D_{lb}(\widetilde{X}, Q)$ is computed as $\sqrt{\sum_{i=1}^{t} lb_i^2}$ according to Theorem 2, 3, 4 and 5.

Algorithm 2: $ESTIMATE\_R_X(c_1, c_2, ..., c_t)$

Input: The cells to which the segments of X belong, $c_1, c_2, ..., c_t$.
Output: $R_X = [\mu_X^{min}, \mu_X^{max}] \times [\sigma_X^{min}, \sigma_X^{max}]$, such that $(\mu_X, \sigma_X) \in R_X$.

1 **for** $1 \le i \le t$ **do**
2 $\quad \mu_i^{min} \leftarrow R(c_i).\mu\_min, \mu_i^{max} \leftarrow R(c_i).\mu\_max$;
3 $\quad \sigma_i^{min} \leftarrow R(c_i).\sigma\_min, \sigma_i^{max} \leftarrow R(c_i).\sigma\_min$;
4 **if** $|Q| \mod w \neq 0$ **then**
5 $\quad \mu_{t+1}^{min} \leftarrow R^{sur}(t).\mu\_min, \mu_{t+1}^{max} \leftarrow R^{sur}(t).\mu\_max$;
6 $\quad \sigma_{t+1}^{min} \leftarrow R^{sur}(t).\sigma\_min, \sigma_{t+1}^{max} \leftarrow R^{sur}(t).\sigma\_min$;
7 $\quad t \leftarrow t + 1$;
8 $sum \leftarrow 0$;
9 **for** $1 \le i \le \lfloor \frac{t}{2} \rfloor$ **do**
10 $\quad$ **if** $\mu_i^{min} > \mu_{t-i}^{max}$ **then**
11 $\quad\quad o' \leftarrow \frac{w_i}{w_i + w_{t-i}} * \mu_i^{min} + \frac{w_{t-i}}{w_i + w_{t-i}} * \mu_{t-i}^{max}$;
12 $\quad\quad sum \leftarrow sum + \frac{w_i}{m}(o' - \mu_i^{min})^2 + \frac{w_{t-i}}{m}(o' - \mu_{t-i}^{max})^2$;
13 $\sigma_X^{min} \leftarrow \sqrt{sum + \sum_{i=1}^{t} \frac{w_i}{m} * \sigma_i^{min2}}$
14 $o \leftarrow \sum_{i=1}^{t} \frac{w_i}{m} * \mu_i^{min} + \sum_{i=1}^{t} \frac{w_i}{m} * \mu_i^{max}$;
15 $\sigma_X^{max} \leftarrow \sum_{i=1}^{t} \frac{w_i}{m} * max\{|o - \mu_i^{min}|, |o - \mu_i^{max}|\}^2$;
16 $\sigma_X^{max} \leftarrow \sqrt{\sigma_X^{max} + \sum_{i=1}^{t} \frac{w_i}{m} * \sigma_i^{max2}}$;
17 $\mu_X^{max} \leftarrow \sum_{i=1}^{t} \frac{w_i}{m} * \mu_i^{max}, \mu_X^{min} \leftarrow \sum_{i=1}^{t} \frac{w_i}{m} * \mu_i^{min}$;
18 **return** $R_X = [\mu_X^{min}, \mu_X^{max}] \times [\sigma_X^{min}, \sigma_X^{max}]$

Phase II involves an extra pruning procedure for *CNED* and *CNDTW* measurements in Lines 15-21. Note that CS_i contains the IDs of cells to which the ith segment of X belongs. For each candidate subsequence X, a range R_X is computed by Algorithm 2 such that $(\mu_X, \sigma_X) \in R_X$. Then, a lower bound of $D(X, Q)$ is computed using $R_X \cap R_{cons}$ rather than R_{cons} as the constraint of normalization in Lines 16-19. In order to distinguish from the lower bound computed with the original constraint R_{cons}, we use $D_{lb}(\widetilde{X}, Q, R_X \cap R_{cons})$ to denote the lower bound function using $R_X \cap R_{cons}$ as the constraint.

Here we explain intuitively why we substitute R_{cons} with $R_X \cap R_{cons}$. Firstly, if $X \in ANS(S, Q, \epsilon)$, then $(\mu_X, \sigma_X) \in R_X \cap R_{cons}$, that is, replacing R_{cons} with $R_X \cap R_{cons}$ still leads to a correct lower bound. Secondly, it is helpful to the efficiency, because $D_{lb}(\widetilde{X}, Q, R_X \cap R_{cons}) \ge D_{lb}(\widetilde{X}, Q)$. Recall the concrete form of $D_{lb}(\widetilde{X}, Q)$ in Theorem 3 and Theorem 5, we can see that a tighter R_{cons} always indicates a tighter lower bound $D_{lb}(\widetilde{X}, Q)$. R_{cons} can be seen as a part of input and could be big. Fortunately, the size of R_X is determined only by the index, and it is much smaller than that of R_{cons}. Therefore, we substitute R_{cons} with $R_X \cap R_{cons}$ to improve the performance of the algorithm.

Phase III. Verification (Lines 22-24). The result is obtained by comparing $D(X, Q)$ with ϵ for each surviving candidate X.

4.3 Analysis of Algorithms

4.3.1 Correctness Analysis. First, we prove $(\mu_X, \sigma_X) \in R_X$ by the correctness of Algorithm 2. The proof of Lemma 4 is omitted [2].

Lemma 4 (correctness of algorithm 2). *If X is completely decomposed into t+1 continuous disjoint segments $\{X^{w_i}_{p_i}\}$, and the first t segments belongs to cell $c_1, c_2, ..., c_t$ respectively, then $(\mu_X, \sigma_X) \in R_X$, which is the output of $ESTIMATE\_R_X(c_1, c_2, ..., c_t)$.*

Theorem 6 (correctness of subsequence matching algorithm). *Algorithm 1 outputs $ANS = ANS(S, Q, \epsilon)$.*

Proof. $\forall X \in ANS(S, Q, \epsilon)$, X will neither be discarded by filtering condition in phase I, nor it will be pruned by $D_{lb}(\widetilde{X}, Q)$ or $D_{lb}(\widetilde{X}, Q, R_X \cap R_{cons})$ in phase II. Then we have $X \in ANS$. Besides, $D(X, Q) \le \epsilon$ for any $X \in ANS$. Thus $X \in ANS(S, Q, \epsilon)$ and $ANS = ANS(S, Q, \epsilon)$. □

4.3.2 Time Complexity. Constructing index for input S requires $O(w|S|)$ time. Subsequence matching queries require $O(|S|\ |Q|)$ time for *ED* or *CNED* and $O(|S||Q|^2)$ time for *DTW* or *CNDTW*.

4.4 Optimize the Segmentation of Q

Since *EPAA* method proposed in Section 3 does not require the lengths of segments to be equal, we can extend the proposed algorithm to exploit multiple indexes.

Assuming there are three indexes built on different segment lengths, whose w are 40, 80 and 160. Then Q can be divided into segments whose length are 40, 80 or 160.

The segmentation of Q effects the size of candidate set and thereby effects the efficiency of our algorithm. Therefore, we improve the efficiency of our algorithm by generating multiple high-quality segmentation plans with heuristic method. Then the cost of query is evaluated heuristically by

$$COST(plan) = \sum_{i=1}^{t} |CS_i| + \gamma \times |S|(\prod_{i=1}^{t} \frac{|CS_i|}{|S| - |Q|})^{\frac{1}{0.5+0.5|S|}}, \quad (14)$$

where $|CS_i| = \sum_{c \in C_i} |SEG_c|$ and γ is a constant obtained from experimental results, which is the ratio between the cost of verifying a candidate and the cost of fetching it from the index. Then, the plan with minimum cost is selected.

5 EXPERIMENTS

5.1 Experiment Setup

Algorithms. We employ two state-of-art algorithms for *subsequence matching problem*, UCR-suit and KV-match, to compare with the proposed algorithm. UCR-suit is an index-free algorithm designed for z-normalized queries, and it needs to scan the whole

[2]The proof can be found in the extended version of this paper at https://gitee.com/su_wen_chang/subsequence_matching.

data sequence. UCR-suit significantly speeds up the query processing by carefully designed lower bound functions to avoid calculating the distance. KV-Match is an index-based algorithm that supports *ED*, *DTW* *CNED* and *CNDTW* with *PAA* summarization method. All these algorithms are implemented in C or C++ and are compiled with g++ 5.4.0 with -O2 level optimization.

Datasets. Our experiments involve both real and synthetic data. The real data used in the following experiments is obtained by concatenating all the series in UCR dataset[3]. The synthetic data is a 10^9-length series and is generated by standard wiener process with restart for every 10^7 elements, which is usually used to simulate the price fluctuations in financial scenarios.

Hardware Environment. The experiments are conducted on a server powered by ubuntu 18.04 with Intel i7 CPUs @2.90GHz, 128GB memory, and 2TB HDD storage.

5.2 Tightness of *EPAA* Summarization Method

This subsection compares the proposed *EPAA* method with *PAA* method in terms of the quality of lower bound functions. The quality of lower bound is measured as the ratio between the estimated lower bound and the precise distance, which is known as the tightness of lower bound (TLB) [13, 27]. A tighter lower bound is helpful to improve the efficiency of algorithm by reducing the candidates to be verified.

Figure 5 compares the tightness of *EPAA* and *PAA* on different distance measurements and query lengths. For each distance measurement, 1000 subsequences are randomly selected from the series S for each length l, where S is the concatenated *UCR* series mentioned above. These subsequences are used in the experiments as queries. We set $l \in \{32, 64, 128, 256, 512\}$ for *CNED* and *CNDTW*, and set $l \in \{32, 64, 128, 256, 512, 1024\}$ for *ED* and *DTW*. The tightness of queries with length l is reported as

$$reported\_TLB = \frac{1}{1000 * (|S| - l + 1)} \sum_{i=1}^{1000} \sum_{j=1}^{|S|-l+1} \frac{D_{lb}(S_j^l, Q_i)}{D(S_j^l, Q_i)},$$

in Figure 5.

Note that for each segment, *EPAA* records both mean and standard deviation while *PAA* only records the mean. Therefore, the space overhead of *EPAA* is actually twice as much as that of *PAA*. To be fair, we use *PAA2* to denote the distance lower bound that is computed using twice as many segments as *EPAA*. Thus, *PAA2* have the same space overhead compared with *EPAA*.

Figure 5(a)~(f) shows the effect of query length over the tightness of *EPAA* and *PAA*. It is clear that the lower bound of *EPAA* is always tighter than that of *PAA* on average. There is no significant difference between the tightness of *EPAA* and the tightness of *PAA2* on *ED* and *DTW* measurements. Besides, the lower bound of *EPAA* is tighter than that of *PAA2* under *CNED* and *CNDTW* measurements. The reason is that *EPAA* contains the standard deviation of each segment and is able to estimate R_X. Therefore, *EPAA* method is a competitive choice for data summarization in *subsequence matching problem*.

Since it is harder to preserve sufficient information for a longer query with the fixed size summarization, the tightness of all the lower bounds decreases while query length increases.

[3]https://www.cs.ucr.edu/~eamonn/time_series_data, accessed on March 1, 2022.

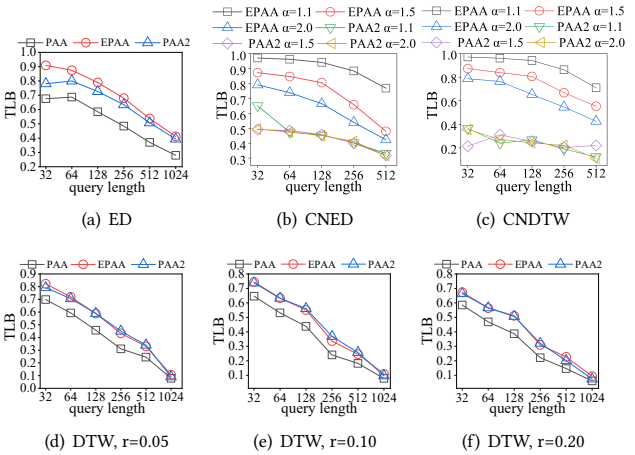


Figure 5: Tightness of *EPAA* and *PAA* methods.

5.3 Subsequence Matching under ED and DTW

This subsection compares the proposed algorithm with KV-Match in terms of query processing time under *ED* and *DTW*. The experiments are performed on synthetic data series. Given S and Q, the selectivity of query is defined as $\frac{|ANS(S,Q,\epsilon)|}{|S|-|Q|}$. Smaller selectivity means smaller value of ϵ.

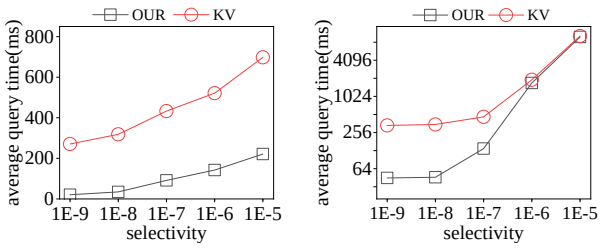
Figure 6 shows the average query time under *ED* and *DTW*. For each selectivity, we randomly select 100 subsequences as the queries from the synthetic data series for each query length $l \in \{128, 256, 512, 1024, 2048\}$ respectively. The average query times for different selectivities are reported in Figure 6(a) and Figure 6(b). The results show that our algorithm is faster than KV-match under *ED* by about 2 to 12 times and is up to 7 times faster under *DTW* when the selectivity of the query is low. However, the proposed algorithm is just slightly better than KV-Match for *DTW* queries with higher selectivities.

The reason of our algorithm being faster can be attributed to the proposed cascading filter strategy. We calculate the proposed $D_{lb}(\widetilde{X}, Q)$ in line 11 of Algorithm 1 while KV-Match does not. Besides, the relative improvement on *DTW* is smaller than *ED* since $DTW_{lb}(\widetilde{X}_{p_i}^{w_i}, U_{p_i}^{w_i}, L_{p_i}^{w_i})$ is more likely to be zero than $ED_{lb}(\widetilde{X}_{p_i}^{w_i}, Q_{p_i}^{w_i})$. Thus, further calculating $D_{lb}(\widetilde{X}, Q)$ for *DTW* is not as helpful as that of *ED* in terms of candidate pruning.

5.4 Subsequence Matching under CNED and CNDTW

Now we compare the efficiency of algorithms under *CNED* and *CNDTW* measurements. In this subsection, UCR-suit, KV-Match and the proposed algorithm are evaluated on the synthetic data. We set query length $l \in \{128, 256, 512, 1024, 2048\}$, set $\alpha \in \{1.1, 1.5, 2.0\}$, $\beta' \in \{0.05, 0.1, 0.2\}$ and $\beta = \beta' \times$ the *Interquartile Range* [4] of $\{\mu_{S_i^w}\}$. Then, R_{cons} is set to $[\mu_Q - \beta, \mu_Q + \beta] \times [\frac{\sigma_Q}{\alpha}, \sigma_Q \times \alpha]$. 25 queries are randomly selected from S for each combination of l, α and β'. Therefore, we have 1125 queries for each selectivity.

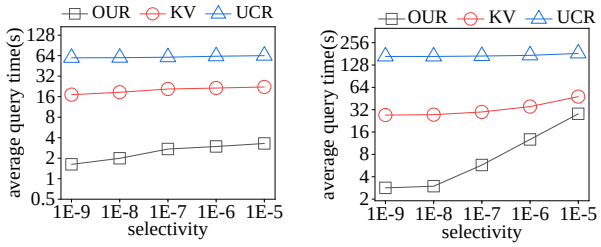
[4]Interquartile Range is defined as the difference between the first and the last quartile.



(a) ED (b) DTW

Figure 6: Average query time under *ED* and *DTW*.

As shown in Figure 7, our algorithm is faster than KV-Match by 3 to 10 times under *CNDTW* and by 3 to 6 times faster under *CNED*. The reason can be explained by the superiority of *EPAA* method over *PAA* on *CNED* and *CNDTW*, which has been evaluated experimentally in subsection 5.2.

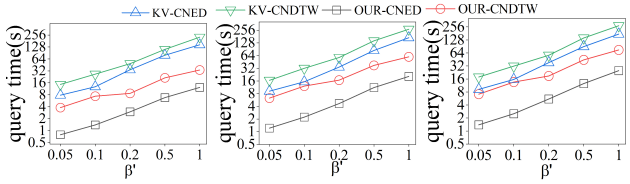


(a) CNED (b) CNDTW

Figure 7: Average query time under *CNED* and *CNDTW*.

5.5 Influence of α and β

This subsection evaluates the influence of α and β, which are the parameters of R_{cons} in Figure 8. It can be seen that the query time goes up with the increase of α and β. The reason is that with bigger α and β, R_{cons} become more loose, and it is harder to calculate the lower bound for both *PAA* and *EPAA* method. The proposed algorithm is always better in each combination of α and β due to the use of *EPAA* method, which is helpful to reduce the effect of large R_{cons} and produces tighter lower bound for *CNED* and *CNDTW*. This is consistent with the result of subsection 5.2.

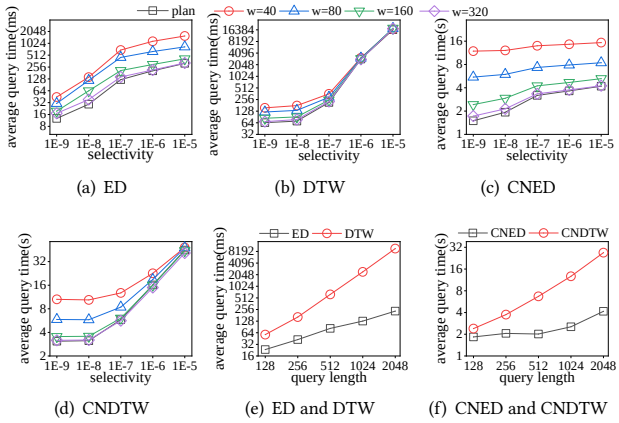


(a) $\alpha = 1.1$ (b) $\alpha = 1.5$ (c) $\alpha = 2.0$

Figure 8: Average query time under different α and β.

5.6 Influence of w and $|Q|$

The effect of the segments length of index w, and the length of query series Q on the performance of the proposed algorithm is illustrated as Figure 9. "plan" stands for the optimized segmentation method discussed in subsection 4.4. Overall, the performance seems to be better with larger value of w. The reason is that $|Q|$ should be bigger than any w involved in this experiment, which unintentionally does favor to longer segments. Besides, the proposed segmentation method have the best performance since it leverages segments of varies lengths.



(a) ED (b) DTW (c) CNED

(d) CNDTW (e) ED and DTW (f) CNED and CNDTW

Figure 9: Average query time vs w and $|Q|$

5.7 Influence of the Index

As is stated before, the proposed algorithm uses a gird index to find the sells that overlap with the query region. This subsection compares our index with the other two competitors. The first one is the grid index that uses equal-width histogram on $\{\mu_{S_i^w}\}$ and $\{\sigma_{S_i^w}\}$ to divide the cells, which is denoted as *Even*. The second one is a quad tree denoted by *Quad*. Our index is denoted as *Quantile*. Besides, *Optimal* reports the minimum number of segments to fetch. The performance of the index is measured by the number of segments fetched into the memory, since the number of candidate segments directly affects the performance of the algorithm.

Table 2 reports the number of the segments fetched into the memory in different index by the average value and standard deviation. There is no significant difference between the average number of the segments fetched into the memory by *Quantile* and that of *Quad*. However, *Quantile* reports smaller standard deviation by about 10% compared with *Quad*, which indicates the degree of performance fluctuation of the proposed grid index is smaller. The reason is that *Quantile* has balanced the number of segments among the rows and columns of the grid while *Quad* only guarantees that the number in each cell does not exceed a certain limit. Besides, *Quad* uses 30% more cells and requires extra overhead to maintain the tree structure. Therefore, it has no superiority over the proposed method. Finally, it is not surprising that *Even* is the worst index, since it is totally unaware of the distribution of the segments.

In a nutshell, the design of our index is reasonable.

Table 2: Number of Segments Fetched by Different Indexes

Method \ Selectitiy		10^{-9}	10^{-8}	10^{-7}	10^{-6}	10^{-5}
Quantile	avg(10^3)	122.4	240.8	629.0	867.0	1078.8
(32768 cells)	std(10^3)	61.4	134.7	349.5	497.6	623.0
Even	avg(10^3)	147.8	270.1	672.7	926.5	1110.7
(32768 cells)	std(10^3)	135.5	228.7	444.4	602.0	724.7
Quad	avg(10^3)	119.2	233.2	628.4	877.2	1065.5
(44744 cells)	std(10^3)	77.9	154.5	392.2	525.5	693.0
Optimal	avg(10^3)	32.0	122.1	493.0	748.8	944.9

6 CONCLUSION

In this work, it is proved that the inherent time complexity of the *subsequence matching problem* is $\omega(n^{1-O(1)})$ even if with the help of any practicable preprocessing. A new summarization method *EPAA* for series data is proposed, which is able to support the distance lower bound for *ED*, *DTW*, *CNED* and *CNDTW*. Based on *EPAA*, an algorithm for solving the subsequence matching problem is designed. The experimental results show that the proposed algorithm is significantly faster then the state-of-art algorithms.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (NSFC) Grant NOs. 61832003, 61972110, 61632010, U1811461, U19A2059, 61872105 and 62072136 and the grant 2019YFB 2101902 and 2020YFB1710200 of National Key R&D Program of China. Thanks Dr. Haoyang Liu for discussing about the ideas. Thanks Zhixin Qi, Xiangyv Gao, Tianpeng Gao and Yifei Li for proof reading the paper.

APPENDIX

This appendix analyzes the time inherent complexity of the subsequence matching problem, the conclusion is as stated in Theorem 1.

Definition 2. *[Nearest Subsequence Problem] Given a query series Q, a long series S and distance measurement D, the nearest subsequence problem is to find a subsequence X^* in S such that* $D(X^*, Q) = \min_{1 \le i \le |S|-|Q|+1} \{D(X_i^{|Q|}, Q)\}$.

Definition 3. *[Approximate Nearest Neighbor problem] Let $E \subset \mathbb{R}^d$. Given a set of d-dimensional vectors, $P = \{p_i | p_i \in E, 1 \le i \le n\}$, a query vector q in E and $\epsilon > 0$, the approximate nearest neighbor problem is to find a vector $p \in P$ such that $D(p, q) \le (1+\epsilon)D(p^*, q)$, where $p^* \in P$ and $D(p^*, q) = \min_{p \in P}\{D(p, q)\}$.*

In the rest of the appendix, we use $NS(S, Q)$ to denote the *nearest subsequence problem* with input (S, Q) and use ϵ-$NN(P, q, \epsilon)$ to denote the *Approximate Nearest Neighbor problem* with input P, q and ϵ. Besides, $S[i]$ refers to the ith element of series S.

Lemma 5. [21] *Assuming Strong Exponential Time Hypothesis (SETH) is true, there exists a constant $\epsilon = \epsilon(\delta, c)$ such that ϵ-$NN(P, q, \epsilon)$ is not computable under L_p for any $p > 0$ in $O(n^{(1-\delta)})$ time for any constants $\delta > 0$ and $c > 0$ even if $O(n^c)$ time preprocessing is allowed, where $n = |P|$.*

Lemma 6. *Given a data series S and a query series Q, the nearest subsequence problem is not computable in time $O(|S|^{1-\delta})$ under L_p distance for any $p > 0$ even if S has been preprocessed in time $O(|S|^c)$ for any constant $c > 1$ and $0 < \delta < 1$ if SETH is true.*

Proof. We prove the lemma by contradiction. Assume that there exists algorithm $\mathcal{A}$ that can solve the $NS(S, Q)$ in time $O(|S|^{1-\delta})$ with or without $O(|S|^c)$ preprocessing. We can design an algorithm $\mathcal{A}'$ from $\mathcal{A}$ to solve ϵ-NN in time $O(|S|^{1-\delta'})$, which contradicts to Lemma 5. Since a vector and a series with limited length are identical in nature, we will take them as the same in the following proof.

Firstly, we show how to solve ϵ-$NN(P, q, \epsilon)$ according to algorithm for $NS(S, Q)$. Assuming there exists an algorithm $\mathcal{A}(\Theta(S), Q)$ for solving $NS(S, Q)$, which outputs X^* in $O(|S|^{1-\delta})$ time after a $O(|S|^c)$ time preprocessing procedure Θ that transforms S to $\Theta(S)$, where c and δ are constants, $c > 1$ and $0 < \delta < 1$. Based on the above assumptions, we can design a preprocessing procedure Θ' for P of ϵ-NN, and an algorithm for ϵ-NN problem $\mathcal{A}'(\Theta'(P), q)$ as follows.

The Θ' is designed from Θ in the following two steps.

Step 1. Transforms P to a series S. Let $S_{pos} = (a, b^1, a, b^2, a, b^3, ..., a, b^{3d+4})$, where $a > \max\{t | t \in p_i, p_i \in P\}$ and $b < \min\{t | t \in p_i, p_i \in P\}$, b^l consists of repeated b and $|b^l| = l$, and d is the same as in Definition 3. We require that a is big enough and b is small enough such that $b < t < a$ for any $t \in \{t | t \in q, q \in E\}$. S is constructed as $(p_1 \circ S_{pos} \circ p_2 \circ S_{pos} \circ ... \circ p_n \circ S_{pos})$, where $X \circ Y$ denotes the concatenation operation of series X and Y. For example, if $X = (1, 2, 3)$ and $Y = (4, 5, 6)$, then $X \circ Y = (1, 2, 3, 4, 5, 6)$.

Step 2. Let $\Theta'(P) = \Theta(S)$.

The algorithm $\mathcal{A}'(\Theta'(P), q)$ for solving ϵ-NN(P, q) works in the following steps.

Step 1. Let $Q = (q \circ S_{pos})$.

Step 2. Compute X^*=$\mathcal{A}(\Theta'(P), Q)$, and outputs $X^*_1{}^d$.

The output of algorithm $\mathcal{A}'$ is the solution to ϵ-$NN(P, q, \epsilon)$, which will be proved in Lemma 7.

Now we show that the existence of Θ' and $\mathcal{A}'$ is contradict to Lemma 5. According to the definition of Θ' and $\mathcal{A}'$, $\Theta'(P)$ cost $O(|S|^c)$ time and $\mathcal{A}'(\Theta'(P), q)$ cost $O(|Q|+|S|^{1-\delta})$ time. For any $d = O(logn)$, we have $|S| = O(n * log^2(n))$ and $|Q| = O(log^2(n))$. When n is big enough, there exists $\delta' < \delta$ such that $(n * log^2(n))^{1-\delta} < (n)^{1-\delta'}$.

Consequently, there exists a constant $\delta' > 0$ such that ϵ-$NN(P, q, \epsilon)$ is computable in $O(n^{1-\delta'})$ time for any $\epsilon > 0$, which contradicts to Lemma 5 if SETH is true. Thus, algorithm $\mathcal{A}$ does not exist. □

Although the proof of Lemma 6 assumes $|Q| = O(log^2 n)$, please note that the conclusion is also true for *ED* when $|Q| = 2^{O(log^* n)}$. This result can be obtained by replacing ϵ-*NN* with *Exact Max-IP Problem over Integers*, whose hardness is given by [4].

Lemma 7. *Algorithm $\mathcal{A}(\Theta'(P), q)$ gives the correct solution to ϵ-$NN(P, q, \epsilon)$ if there exists an algorithm $\mathcal{A}(\Theta(S), Q)$ as stated in the proof of Lemma 6.*

Proof. We only need to prove that $X^{*d}_1 \in P$ and $L_p(X^{*d}_1, q) \leq L_p(p, q)$ for any $p \in P$.

$X^{*d}_1 \in P$ is proved as follows.

First, we prove *proposition* 1, that is, $L_p(X, Q) < \sqrt[p]{d}(a-b)$ if $X^d_1 \in P$ for any $|Q|$-length subsequence X. Since $X^d_1 \in P$, $X^{|Q|-d}_{d+1} = S_{pos} = Q^{|Q|-d}_{d+1}$. Therefore,

$$L_p(X, Q) = (\sum_{i=1}^{d} |x_i - q_i|^p + \sum_{i=d+1}^{|Q|} |x_i - q_i|^p)^{\frac{1}{p}} = L_p(X^d_1, q).$$

From $|x_i - q_i| < a - b$, $L_p(X^d_1, q) < \sqrt[p]{d}(a-b)$, we have $L_p(X, Q) < \sqrt[p]{d}(a-b)$.

Then, we prove *proposition* 2, that is, $L_p(X, Q) \geq \sqrt[p]{d}(a-b)$ if $X^d_1 \notin P$ for any $|Q|$-length subsequence X. Let ${S_{pos}}_i^w$ and ${S_{pos}}_j^w$ be two different subsequences of S_{pos}, $K = \{k | {S_{pos}}_i^w[k] = {S_{pos}}_j^w[k] = a\} = \{k_1, k_2, ...k_{|K|}\}$, and $K' = \{k' | {S_{pos}}_i^w[k'] \neq {S_{pos}}_j^w[k']\}$. Without loss of generality, we assume $i < j$ and $k_1 < k_2 < ... < k_{|K|}$. By the definition of L_p, we have

$$\begin{aligned} L_p({S_{pos}}_i^w, {S_{pos}}_j^w) &= (\sum_{k'_l \in K'} |{S_{pos}}_i^w[k'_l] - {S_{pos}}_j^w[k'_l]|^p)^{\frac{1}{p}} \\ &= \sqrt[p]{|K'|}(a-b). \end{aligned} \tag{15}$$

$\forall k_l, k_{l+1} \in K$, there must exist $k'_l \in K'$ such that $k_l < k'_l < k_{l+1}$. Otherwise ${S_{pos}}^{k_{l+1}-k_l}_{i+k_l} = {S_{pos}}^{k_{l+1}-k_l}_{j+k_l} = (a, b^{k_{l+1}-k_l-1}, a)$, which implies $i = j$ because the number of "b"s between two different pairs of "a"s is different in S_{pos}. Consequently, $|K'| \geq |K| - 1$. Let $CntA(S)$ denote the number of "a"s in series S. Since every "a" in series ${S_{pos}}_i^w$ contributes one element to either K or K', $|K'| + |K| \geq CntA({S_{pos}}_i^w)$. Therefore,

$$L_p({S_{pos}}_i^w, {S_{pos}}_j^w) \geq (\lfloor \frac{CntA({S_{pos}}_i^w) - 1}{2} \rfloor)^{\frac{1}{p}} * (a-b) \tag{16}$$

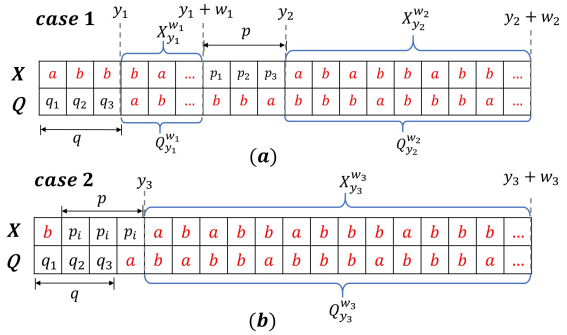


Figure 10: Two cases for $X^d_1 \notin P$. The elements from S_{pos} are painted in red. a)$\forall r \in \{1, 2...d\}, X[r] = a$ or $X[r] = b$. b)$\exists r \in \{1, 2...d\}$ that $X[r] \neq a$ and $X[r] \neq b$.

If $X^d_1 \notin P$, there exists two cases as illustrated in Figure 10. In the first case shown as Figure 10 (a), $X[r] \neq a$ and $X[r] \neq b$ for any $r \in \{1, 2...d\}$. Therefore, there exist y_1, w_1, i and j such that $X^{w_1}_{y_1} = {S_{pos}}_i^{w_1}$ and $Q^{w_1}_{y_1} = {S_{pos}}_j^{w_1}$. Besides, there also exist y_2, w_2, i' and j' such that $y_2 > y_1 + w$, $X^{w_2}_{y_2} = {S_{pos}}_{i'}^{w_2}$ and $Q^{w_2}_{y_2} = {S_{pos}}_{j'}^{w_2}$. Thus, $w_1 + w_2 \geq |Q| - 2d = \frac{(3d+4)(3d+3)-2d}{2}$. Therefore, $CntA(Q^{w_1}_{y_1}) + CntA(Q^{w_2}_{y_2}) \geq 2d + 4$. According to *inequality (16)*, we have

$$\begin{aligned} L_p(X, Q) &\geq [L_p(X^{w_1}_{y_1}, Q^{w_1}_{y_1})^p + L_p(X^{w_2}_{y_2}, Q^{w_2}_{y_2})^p]^{\frac{1}{p}} \\ &\geq [\frac{CntA(Q^{w_1}_{y_1}) + CntA(Q^{w_2}_{y_2})}{2} - 2]^{\frac{1}{p}} * (a-b) \\ &\geq \sqrt[p]{d}(a-b). \end{aligned} \tag{17}$$

In case 2 as illustrated as Figure 10(b), $\exists r \in \{1, 2...d\}$ such that $X[r] \neq a$ and $X[r] \neq b$. Thus, there exist y_3 and w_3 such that $X^{w_3}_{y_3} = {S_{pos}}_i^{w_3}$ and $CntA(Q^{w_3}_{y_3}) \geq 2d + 4$. Therefore, $L_p(X, Q) \geq \sqrt[p]{d}(a-b)$ if $X^d_1 \notin P$.

Since X^* is the solution to $NS(S, Q)$, we have $L_p(X^*, Q) < \sqrt[p]{d}(a-b)$ according to *proposition 1*. According to *proposition 2*, $L_p(X, Q) > L_p(X^*, Q)$ for any $X^d_1 \notin P$. Therefore $X^*[1:d] \in P$.

Now, we prove that $\forall p \in P$, $L_p(X^{*d}_1, q) \leq L_p(p, q)$ by contradiction. Assuming $\exists p' \in P$ such that $L_p(X^{*d}_1, q) > L_p(p', q)$, S must contains a $|Q|$-length subsequence $X = (p' \circ S_{pos})$ such that $L_p(X, Q) = L_p(p', q)$. Therefore, $L_p(X, Q) = L_p(p', q) < L_p(X^*, Q)$, which contradicts with the fact that X^* is the solution to $NS(S, Q)$. Therefore, $L_p(X^{*d}_1, q) \leq L_p(p, q)$ for any $p \in P$.

Till now, we have proved $X^{*d}_1 \in P$ and $L_p(X^{*d}_1, q) \leq L_p(p, q)$ for any $p \in P$, which means that X^{*d}_1 is the solution of ϵ-$NN(P, q, \epsilon)$. □

Finally, the inherent hardness of *subsequence matching problem*, which is Theorem 1 in Section II, is proved as follows.

Proof of Theorem 1. Since the solution of $NS(S, Q)$ is in $ANS(S, Q, \epsilon)$ if $\epsilon \geq L_p(X^*, Q)$, $NS(S, Q)$ can be solved by one scan of $ANS(S, Q, \epsilon_1)$. Denote the time to compute $NS(S, Q)$ as T_1, the time to compute $ANS(S, Q, \epsilon)$ as T_2, and the time to scan $ANS(S, Q, \epsilon)$ as T_3. Therefore, $T_1 \leq T_2 + T_3$. Since $T_2 \leq T_3$, $\frac{1}{2}T1 \leq T2$. According to Lemma 6, $T_2 = \omega(n^{1-\delta})$ for any constant $\delta > 0$. Therefore, Theorem 1 is true. □

Therefore, we have to accept approximate results if we want to find the algorithms with better time complexity, perhaps with the similar method in [3].

REFERENCES

[1] Noura Alghamdi, Liang Zhang, Huayi Zhang, Elke A. Rundensteiner, and Mohamed Y. Eltabakh. 2020. ChainLink: Indexing Big Time Series Data For Long Subsequence Matching. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. 529–540. https://doi.org/10.1109/ICDE48307.2020.00052

[2] Zhipeng Cai and Zaobo He. 2019. Trading private range counting over big IoT data. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 144–153.

[3] Zhipeng Cai, Dongjing Miao, and Yingshu Li. 2019. Deletion propagation for multiple key preserving conjunctive queries: approximations and complexity.

In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 506–517.

[4] Lijie Chen. 2018. On the hardness of approximate and exact (bichromatic) maximum inner product. *arXiv preprint arXiv:1802.02325* (2018).

[5] Lei Chen and Raymond Ng. 2004. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. 792–803.

[6] Lei Chen, M Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. 491–502.

[7] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2020. The lernaean hydra of data series similarity search: An experimental evaluation of the state of the art. *arXiv preprint arXiv:2006.11454* (2020).

[8] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. 1994. Fast Subsequence Matching in Time-Series Databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, Minneapolis, Minnesota, USA, May 24-27, 1994*, Richard T. Snodgrass and Marianne Winslett (Eds.). ACM Press, 419–429. https://doi.org/10.1145/191839.191925

[9] Wook-Shin Han, Jinsoo Lee, Yang-Sae Moon, and Haifeng Jiang. 2007. Ranked subsequence matching in time-series databases. In *Proceedings of the 33rd international conference on Very large data bases*. Citeseer, 423–434.

[10] Yoonho Hwang and Hee-Kap Ahn. 2011. Convergent bounds on the euclidean distance. In *Advances in neural information processing systems*. 388–396.

[11] Russell Impagliazzo and Ramamohan Paturi. 2001. On the complexity of k-SAT. *J. Comput. System Sci.* 62, 2 (2001), 367–375.

[12] Kunio Kashino, Gavin Smith, and Hiroshi Murase. 1999. Time-series active search for quick retrieval of audio and video. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, Vol. 6. IEEE, 2993–2996.

[13] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems* 3, 3 (2001), 263–286.

[14] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowledge and information systems* 7, 3 (2005), 358–386.

[15] Satoshi Koide, Chuan Xiao, and Yoshiharu Ishikawa. 2020. Fast Subtrajectory Similarity Search in Road Networks under Weighted Edit Distance Constraints. (2020).

[16] Michele Linardi and Themis Palpanas. 2020. Scalable Data Series Subsequence Matching with ULISSE. *The VLDB Journal* 8 (2020), 1–26.

[17] Yang-Sae Moon, Kyu-Young Whang, and Wook-Shin Han. 2002. General match: a subsequence matching method in time-series databases based on generalized windows. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, USA, June 3-6, 2002*, Michael J. Franklin, Bongki Moon, and Anastassia Ailamaki (Eds.). ACM, 382–393. https://doi.org/10.1145/564691.564735

[18] Yang-Sae Moon, Kyu-Young Whang, and Woong-Kee Loh. 2001. Duality-Based Subsequence Matching in Time-Series Databases. In *Proceedings of the 17th International Conference on Data Engineering, April 2-6, 2001, Heidelberg, Germany*, Dimitrios Georgakopoulos and Alexander Buchmann (Eds.). IEEE Computer Society, 263–272. https://doi.org/10.1109/ICDE.2001.914837

[19] Thanawin Rakthanmanon, Bilson J. L. Campana, Abdullah Mueen, Gustavo E. A. P. A. Batista, M. Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn J. Keogh. 2012. Searching and mining trillions of time series subsequences under dynamic time warping. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, Qiang Yang, Deepak Agarwal, and Jian Pei (Eds.). ACM, 262–270. https://doi.org/10.1145/2339530.2339576

[20] Sayan Ranu, Padmanabhan Deepak, Aditya D Telang, Prasad Deshpande, and Sriram Raghavan. 2015. Indexing and matching trajectories under inconsistent sampling rates. In *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 999–1010.

[21] Aviad Rubinstein. 2018. Hardness of approximate nearest neighbor search. In *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing*. 1260–1268.

[22] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26, 1 (1978), 43–49.

[23] Dennis Shasha. 1999. Tuning time series queries in finance: Case studies and recommendations. *IEEE Data Eng. Bull.* 22, 2 (1999), 40–46.

[24] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. 2020. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal* 29, 1 (2020), 3–32.

[25] T. Sun, H. Liu, S. Mcloone, S. Ji, and X. Wu. 2020. Time series indexing by dynamic covering with cross-range constraints. *The VLDB Journal* (2020), 1–20.

[26] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Proceedings 18th international conference on data engineering*. IEEE, 673–684.

[27] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. 2013. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery* 26, 2 (2013), 275–309.

[28] Yang Wang, Peng Wang, Jian Pei, Wei Wang, and Sheng Huang. 2013. A data-adaptive and dynamic segmentation index for whole matching on time series. *Proceedings of the Vldb Endowment* 6, 10 (2013), 793–804.

[29] Jiaye Wu, Peng Wang, Ningting Pan, Chen Wang, Wei Wang, and Jianmin Wang. 2019. KV-Match: A Subsequence Matching Approach Supporting Normalization and Time Warping. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*.

[30] Djamel Edine Yagoubi, Reza Akbarinia, Florent Masseglia, and Themis Palpanas. 2017. Dpisax: Massively distributed partitioned isax. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1135–1140.

[31] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn J. Keogh. 2016. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, Francesco Bonchi, Josep Domingo-Ferrer, Ricardo Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu (Eds.). IEEE Computer Society, 1317–1322. https://doi.org/10.1109/ICDM.2016.0179

[32] Yan Zhu, Makoto Imamura, Daniel Nikovski, and Eamonn J. Keogh. 2017. Matrix Profile VII: Time Series Chains: A New Primitive for Time Series Data Mining (Best Student Paper Award). In *2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017*, Vijay Raghavan, Srinivas Aluru, George Karypis, Lucio Miele, and Xindong Wu (Eds.). IEEE Computer Society, 695–704. https://doi.org/10.1109/ICDM.2017.79

[33] Yunyue Zhu and Dennis Shasha. 2003. Warping indexes with envelope transforms for query by humming. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. 181–192.

[34] Kostas Zoumpatianos and Themis Palpanas. 2018. Data series management: Fulfilling the need for big sequence analytics. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1677–1678.